



WWW.MICROSIZEME.at

Allgemein

Anschrift

HTL Ottakring
Thaliastraße 125 1160 Wien
Tel.: 01 49111-113

Fax: 01 49111-199

Projektteam

DANILIUC Andreas 5BHITN
HAJI HASHEMI Sasan 5BHITN
PROCHAZKA Simon 5BHITN
SPANDL Jürgen 5BHITN
SUŠKO Patrik 5BHITN

Projektleiter

HAJI HASHEMI Sasan
e-Mail: sasan.hajihashemi@hotmail.com
Mobil: 0699/17207886

Projektbetreuer

Prof. DI Robert Baumgartner MBA
e-Mail: lbaum@htl-ottakring.at

Projekttitel: MicrosizeMe

Projektnummer: J110287

Kategorie: Sonderpreis IKT

Spezifikation: Informations-/Kommunikationstechnik

Inhaltsverzeichnis

Allgemein.....	2
Projektentstehung- und Planung.....	4
Entstehung der Idee	4
Ziel des Projekts	4
Termine	5
Meilensteine	5
Aufgabenverteilung.....	6
Ressourcenplanung.....	7
Recherchen	8
Inhaltliche Beschreibung des Projekts	9
Was ist MicrosizeMe?	9
Entwicklung Modellierung.....	10
Entwicklung Animation.....	12
Entwicklung Singleplayer Scripting.....	13
Entwicklung Singleplayer Scripting.....	13
Marketing	14
Entwicklung Multiplayer	15
Bericht der Projektkoordinator/In.....	17
Zusammenarbeit innerhalb des Teams.....	17
Zusammenarbeit zwischen Team und Projektbetreuer	17
Zusammenarbeit zwischen Team und Schule.....	17
Lernen im Team	18
Diskussionen und Konflikte	18
Anhang	19

Projektentstehung- und Planung

Entstehung der Idee

Die Idee entstand an einem Tag der offenen Tür. Bei einer Diskussion mit Mitschülern kamen wir zu dem Ergebnis, dass die Präsentation der Schule durch den Tag der offenen Tür und die Schulhomepage aus unserer Sicht nicht ausreichend abgedeckt ist. Auf einen Unterstufenabsolventen kann diese Masse an Information unübersichtlich und trocken wirken. Also überlegten wir uns einen innovativen Weg unsere Schule zu präsentieren. Spezielle Zielgruppe des Projektes war die Jugend. Wir entschieden uns dafür, ein 3D Spiel zu entwerfen, das einerseits speziell junges Publikum durch dessen spielerischen Aufbau begeistern sollte, andererseits aber auch Fakten über die Schule vermitteln sollte. Der Benutzer sollte sich virtuell frei durch die Schule bewegen können und sich mit dieser vertraut machen, aber auch die wichtigsten Aspekte dieser präsentiert bekommen.

Als wir unser Team gegründet hatten mussten wir uns an einen Lehrer wenden, den wir gerne als unseren betreuenden Lehrer hätten. Die Aufgabe des betreuenden Lehrers ist es, das Projekt zu überwachen und regelmäßig den Fortschritt und Zwischenstand zu prüfen. Herr Prof. Baumgartner erklärte sich bereit die Betreuung unserer Gruppe zu übernehmen und riet uns bereits in den Sommerferien ein Konzept für das Projekt zu erarbeiten

Ziel des Projekts

Ziel des Projektes ist es, mit Hilfe eines 3D Computerspiels welches auf unserer Schulhomepage präsentiert wird, die HTL Ottakring vorzustellen und gleichzeitig Wissenswertes über unsere Schule zu vermitteln, Neugier zu wecken und Interesse an einer technischen Ausbildung zu fördern. Um dieses Erlebnis so realitätsgetreu wie möglich zu gestalten wurde von der Schule ein 3D Model erstellt. Die Forderung, dass das Spiel über den Browser ohne jegliche weiteren Applikationen ausführbar ist, war uns sehr wichtig. So können wir eine größere Zielgruppe für unsere Schule begeistern.

Das Sekundärziel besteht aus der Entwicklung eines Multiplayers in dem sich Spieler über das Web miteinander in Teams gegen gegnerische Teams beweisen können.

Termine

Die Terminplanung war keine einfache Angelegenheit, da wir uns zu Beginn des Projektes sehr unsicher bezüglich des Zeitaufwandes einzelner Teilaufgaben waren. Da das Entwickeln von interaktiven 3D-Umgebungen für uns noch neu war, mussten wir uns sämtliche Fähigkeiten von Grund auf aneignen. Dass hierbei auch ab und zu Fehler passierten merkten wir schon relativ früh. Dementsprechend stellte sich uns auch das Problem mit dem festlegen von Terminen und Fristen: Aufgrund der immer wieder auftretenden Fehler und Probleme mussten wir zusätzliche Zeitpölster einplanen. Das Problem war jedoch, dass der Zeitplan aufgrund der hohen Anzahl an Teilaufgaben ohnehin schon sehr beschränkt war. Es ließen sich lediglich gröbere Termine festlegen, beispielsweise Fristen für den Abschluss einzelner Zwischenlevels. Die einzigen in Stein gemeißelten Termine waren sämtliche Zwischenpräsentationen, die hauptsächlich für den Projektbetreuer, den Abteilungsleiter oder die Landesschulinspektorin festgelegt wurden, aber auch der Abgabetermin der Diplomarbeit inklusive der fertigen Projektdokumentation.

Meilensteine

Nr.	Ereignis	Termin-Soll	Termin-Ist
1	Brainstorming, Namensfindung, Zielsetzung	14.09.2010	14.09.2010
2	Planung fertig gestellt	21.09.2010	28.09.2010
3	Level 1 (Außenbereich) fertig gestellt	09.11.2010	24.11.2010
4	Level 2 (Aula) fertig gestellt	21.01.2011	21.01.2011
5	Level 3 (Lift) fertig gestellt	22.02.2011	22.02.2011
	Multiplayer	01.03.2011	
6	Level 4 (Cafeteria) fertig gestellt	15.03.2011	
7	Level 5 (Klassenzimmer) fertig gestellt	29.03.2011	
8	Diplomarbeitsdokumentation fertig gestellt	12.04.2011	
9	Projektendpräsentation	09.05.2011	

Aufgabenverteilung

Daniliuc Andreas

- Game Concept
- Modelling
- Storyline
- Visual Effects

Haji Hashemi Sasan

- Projektmanagement
- Planung und Koordination
- Game Design
- Marketing

Prochazka Simon

- Game Concept
- Scripting (Singleplayer)
- Scripting (Multiplayer)
- Level Editor

Spandl Jürgen

- Game Concept
- Modellierung
- Character-Animation
- In-Game Audio

Suško Patrik

- Game Concept
- Scripting (Singleplayer)
- Schnittstelle zwischen Cinema 4D und Unity 3D
- Webscripting

Ressourcenplanung

Zeitressourcen

In der Schule stehen uns laut Stundenplan 5h/Woche für das Projekt zur Verfügung. In Anbetracht des Projektumfangs ist das Projekt jedoch nicht in dieser Zeit realisierbar, was zu einem enormen zeitlichen Aufwand außerhalb der Schule führt. Der tatsächliche Stundenaufwand des Projektes bis dato beläuft sich auf geschätzt 100-300 Stunden, wobei der größte Teil der Arbeit außerhalb der Schule erledigt wurde.

Raumressourcen

Wöchentlich ist es uns möglich fünf Stunden lang einen von der Schule zur Verfügung gestellten Raum zu nutzen. Oftmals kommt es auch außerhalb der Schule zu Treffen und Meetings, die entweder in der Bibliothek oder direkt bei Projektmitgliedern zu Hause stattfinden.

Hardware Ressourcen

Grundsätzlich arbeitet jeder Schüler auf seinem eigenen Notebook, falls notwendig werden beispielsweise für aufwändige Berechnungen (z.B. Rendering) auch von der Schule zur Verfügung gestellte Rechner eingesetzt.

Software Ressourcen

Scripting

- Unity 3D Professional

Modelling

- Maxon Cinema 4D
- Blender

Grafisches Design

- Adobe Photoshop CS5

Schriftführung

- Microsoft Office 2010

Recherchen

Unsere Recherchen begannen an dem Zeitpunkt als wir unser Ziel definiert hatten. Wir überlegten welche Software am besten für uns geeignet ist, um ein Spiel von diesem Ausmaß zu entwickeln. Unser Projektbetreuer stellte uns Unity 3D vor, eine Entwicklungsebene die das direkte Programmieren eines Spiels ermöglichen sollte. Nun fehlte uns noch eine Software zum Modellieren der einzelnen 3D Objekte und Umgebungen, sowie zum Erstellen von Animationen. Nach langer Suche entschieden wir uns für die Modellierungssoftware Cinema 4d aus dem Hause Maxon.

Sowohl die Recherche als auch das Aneignen der notwendigen Fähigkeiten konnte nun beginnen. Wir teilten die Entwicklung des Spieles in folgende drei Unterpunkte auf:

Modellierung

Da die Modellierungssoftware eine beinahe unüberschaubare Vielzahl an Funktionen besaß, mussten wir uns erstmals mit ihnen vertraut machen. Nach unzähligen Tutorials und einigen Probeläufen gelang es unseren Modellierern endlich, einen Prototyp des ersten Levels zu erzeugen. Glücklicherweise besitzt Cinema 4D eine sehr starke Community, die uns speziell mit Lehrvideos und schriftlich festgehaltenen Anleitungen und Ratschlag-Gebern zur Seite stand. Wir wendeten quasi die Methode „learning by doing“ an, um uns mit dem Programm vertraut zu machen.

Scripting

Die vier Jahre Programmierunterricht zuvor an der HTL erleichterten die Recherche im Bereich scripting. Außerdem war unsere Gruppe durch Projekte im vierten Jahrgang bereits mit der Unity 3D Software vertraut. Nichtsdestotrotz waren zahlreiche Recherchen besonders in den Bereichen JavaScript-Programmierung und bei Unity 3D spezifischen Funktionen Notwendig. Lehrvideos auf Youtube und die offizielle Unity 3D Homepage dienten uns als Informationsquellen.

Game Design

Um ein spannendes Jump'n'Run-Game zu entwickeln braucht es aber mehr als ein fertiges 3D Modell und die dazu gehörigen Scripts. Diese nutzen nichts, wenn man nicht weiß wie man sie einzusetzen hat. Bei der Recherche bezüglich des Game Designs ging es darum, der virtuellen Welt „Leben“ einzuhauchen und die spielerische sowie die grafische Attraktivität zu steigern. Wir mussten uns also herausfordernde Spielelemente und eine Hintergrundgeschichte ausdenken. Selbstverständlich nahmen wir uns Elemente aus anderen erfolgreichen Spielen zum Vorbild. Wir versuchten, den Stil unseres Projektes nicht zu sehr davon abweichen zu lassen, ohne aber dabei lediglich ein Replikat zu erstellen. Außerdem versuchten wir uns auf ihrer Basis neue innovative

Designs auszudenken. Ein weiteres Ziel war es also quasi, ein Computerspiel von Spielern, für Spieler zu erzeugen.

Inhaltliche Beschreibung des Projekts

Was ist MicrosizeMe?

Vor der Schule wird man auf mysteriöse Weise geschrumpft und man muss sich nun als kleiner Mann einen Weg durch die Schule suchen. Als Spieltyp haben wir uns für „Jump and Run“ entschieden da dieser sich am besten eignet für eine virtuelle Führung durch die Schule. Der spielerische Aspekt bei der Erkundung der Schule war uns sehr wichtig. Mit der Programmierentwicklungsumgebung Unity 3D in Verbindung mit Cinema 4D eine 3D Grafik Software war es uns möglich ein nahezu authentisches Umfeld zu erschaffen welches einen starken wiedererkennungswert besitzt und dadurch den Spieler in seinen Bann zieht.

Die einzelnen Bereiche des Spieles erstrecken sich von der Eingangstreppe über die Aula bis zu Klassenzimmer und fordern durch trickreiche Hindernisse und Minigames die Fingerfertigkeit des Spielers heraus. Die Browserkompatibilität ist durch Unity 3D gewährleistet. In diesem Bereich wollen wir uns mit MicrosizeMe etablieren. Die Interessenten benötigen lediglich einen Internetanschluss.

Dies ermöglicht natürlich große Möglichkeiten im Bereich Marketing. Durch virtuelle Werbeflächen welche aus Grafiken und Videos bestehen bieten wir externen Unternehmen eine Präsentationmöglichkeit oder können die Werbeflächen auch in Sachen Eigenwerbung für die Schule (Veranstaltungen, Schulbuffet etc.) verwenden. So wird zum Beispiel ein Lehrvideo über die Schule in der virtuellen Aula projiziert. Die Kombination aus virtuellem Tag der offenen Tür und einem anspruchsvollen Computerspiel war die Herausforderung der wir uns stellen mussten.

Zusätzlich zum Singleplayer, der den Benutzer durch die Bereiche der Schule führt besteht MicrosizeMe aus einem Multiplayer Modus. Wir wählten den Spielmodi „Capture the Flag“ da es das Spiel automatisch Taktischer, somit auch interessanter gestaltet. Um die Flagge zu erkämpfen bzw. zu beschützen besitzt der Spieler über spezielle Fähigkeit die er im richtigen Moment benutzen muss um sich und seine Teamkameraden zum Sieg zu führen. Das Teamplay unter den Spielern steht beim Multiplayer klar im Vordergrund.

Entwicklung Modellierung

3D Objects

In jedem dreidimensionalen Spiel müssen als erstes die für das Spiel notwendigen Figuren und Objekte modelliert werden. Um eine realistische 3D-Umgebung – beispielsweise einen Raum – erstellen zu können, benötigt man als erstes eine Menge detaillierter Aufnahmen des Raumes. Einerseits wird dadurch die Modellierung selber vereinfacht (da man jedes Modell auch von zu Hause aus betrachten kann ohne es noch einmal „live“ sehen zu müssen), andererseits benötigt man die Fotos als Rohdaten für die Erzeugung der Texturen.

Texturen

Texturen sind sozusagen die Haut eines Objektes. Ein fertiges Modell enthält nur die genaue Form eines Objektes (Abb.1), nicht jedoch die Farbinformation (Bilder, Muster etc.). Deswegen muss jedes fertige Modell noch mit einer Textur (Abb.2) versehen werden, die das sonst grau erscheinende Modell in den entsprechenden Farben einfärbt (Abb.3).

Modellierung Außenbereich der Schule

Die größte Schwierigkeit beim ersten Level war die Gestaltung der Schulumgebung. Da die komplette Modellierung der umliegenden Gebäude monatelange Arbeit bedeutet hätte, musste eine Idee her um die Arbeit zu reduzieren. Wir entschieden uns schlussendlich dafür, ein Panoramabild der Umgebung zu erstellen, und als Textur auf eine modellierte Kuppel zu legen.

Um das Panoramabild zu erstellen wurden aus einer zentral gelegenen Position etwa 60 Einzelbilder im Winkel von 180° aufgenommen und anschließend in Photoshop zu einem Panoramabild zusammengefügt. Dadurch haben wir uns einerseits viel Modellierungsarbeit gespart, andererseits konnten wir auch ein viel realistischeres Bild der Umgebung erzeugen (Abb.4).

Um dem Ganzen noch den letzten Feinschliff zu verpassen haben wir das Modell noch mit einem sogenannten Sky-Objekt versehen. Das Sky-Objekt simuliert einen realitätsgetreuen Himmel mit verstellbarer Wolkenanzahl, der die Gesamtbeleuchtung der Szene etwas aufpeppen soll (Abb.5).

Zusätzlich zu den reell vorhandenen Objekten wurden auch kleinere Objekte dazu modelliert, beispielsweise das Katapult das den Spieler auf eine höher gelegene Plattform wirft (Abb.6) oder die am Geländer herunterrollenden Getränkedosen, denen der Spieler ausweichen muss (Abb.7).

Modellierung Aula

Der zweite Level des Spiels (Abb.8) war bislang modelliertechnisch der aufwändigste Level. Mit der steigenden Anzahl der (speziell kleineren) Objekte stieg auch der Modellierungsaufwand. Die größte Schwierigkeit in dem Level waren wahrscheinlich die bis ins letzte Detail verzierten Säulen (Abb.9) und Wände. Beim dreidimensionalen nachbauen dieser Elemente mussten wir ziemlich genau arbeiten, wodurch unsere Szenen immer komplexer und rechenaufwändiger wurden.

Um das Level etwas dynamischer zu machen wurden zwischen den Säulen Seile gespannt, auf denen der Spieler von einem Punkt zum anderen balancieren muss. Dadurch steigt nicht nur der Spaßfaktor im Spiel, dem Spieler wird auch ein guter Ausblick auf den Level ermöglicht.

Ein weiterer wichtiger Teil des Levels ist das Minigame am Ende des Levels. Um zum nächsten Level zu kommen muss der Spieler zuerst mit einem Fahrzeug (Tron Lightcycle, ein Fahrzeug aus dem gleichnamigen Film) eine Rennstrecke absolvieren (Abb.3). Das Rennen ist unterteilt in einzelne Checkpoints, die nacheinander abgefahren werden müssen. Teilweise sind die Checkpoints des Rennens so platziert, dass man sich geschickt zwischen den Stuhl- und Tischbeinen hindurchschlängeln muss. Am Ende des Rennens bekommt der Spieler den Aufzugsschlüssel, der schlussendlich für die Inbetriebnahme des Aufzuges notwendig ist.

Modellierung Aufzug

Der dritte Level des Spiels spielt sich ausschließlich im Aufzug ab. Dadurch dass der Spieler geschrumpft ist, muss er irgendwie an die Knöpfe kommen, die zur Inbetriebnahme des Aufzuges gedrückt werden müssen. Um nun an die Knöpfe zu kommen haben wir uns ein weiteres kleines Minigame überlegt, in dem der Spieler sich im klassischen Tetris-Stil eine Treppe nach oben bauen muss, und diese stufenweise nach oben gehen muss.

Wenn der Knopf zur ersten Etage gedrückt ist, setzt sich der Aufzug in Bewegung und spielt fährt nach oben. Wenn er wieder zum Stehen kommt ist der Level beendet.

Um das Aussehen des Aufzuges etwas zu verschönern wurden neben realistischen Metalltexturen für die Wände (es handelt sich um einen Aluminiumaufzug) auch Spiegelungseffekte und dynamische Leuchteffekte hinzugefügt.

Modellierung Cafeteria

Der vierte Level unseres Spiels spielt sich in der Schulcafeteria ab. Da in der Cafeteria sehr viele Tische stehen, haben wir diese in unser Spielkonzept einbezogen und ein Skateboard-Level daraus gemacht. Ziel ist es, von einem Tisch immer wieder über verschiedene Rampen auf den nächsten Tisch zu kommen.

Die Rampenmodellierung und kleinere Objekte wie Flaschen und Dosen sind der Hauptbestandteil der Modellierungsarbeit in diesem Level.

Entwicklung Animation

Figur - Animation

Die Animation der Spielfigur spielt eine wesentliche Rolle in MicrosizeMe. Hierfür wird das Modul Mocca der 3D – Grafik – Software Cinema 4D verwendet. Das Mocca Modul stellt wichtige Werkzeuge und Funktionen für die Realisierung von (Charakter-)Animationen zur Verfügung.

Rigging

Nachdem eine Spielfigur fertig modelliert und texturiert ist, ist das Rigging der erste Schritt der getan wird um unseren Charakter einer Animation näher zu bringen. Zu Beginn wird ein virtuelles Skelett aus gelenkähnlichen Objekten, den Joints, gebildet. Ein Joint beschreibt die Position und die Achsen - Ausrichtung im Raum. Zwischen den Joints werden Bones gebildet, welche Einfluss auf die Deformation des zu animierenden Drahtgittermodells haben. Dieses Skelett ist oft an die Anatomie eines echten Skeletts angelehnt.

Skinning

Nachdem die Figur fertig geriggt wurde, folgt als nächster Schritt das Skinning. Hierbei wird das zu animierende Drahtgittermodell dem fertigen Skelett zugeteilt und darauf gebunden. Ab diesem Schritt kann durch eine Veränderung der Joint Koordinaten eine Bewegung erzeugt werden.

Gewichtung

Durch das Ausführen des Skinning Prozesses wird das Modell automatisch gewichtet. Das heißt, dass den einzelnen Joints zugeteilt wird, welche Teile der Figur sich wie stark bewegen und deformieren, wenn ihre Einstellungen verändert werden. Die automatische Gewichtung erfüllt in vielen Fällen jedoch nicht die gestellten Anforderungen und so muss der Animator oft mithilfe eines Wichtungswerkzeuges, durch aufmalen und radieren, nachbessern. Eine gute Gewichtung ist ein sehr wesentlicher Punkt einer gelungenen Animation. Wird hierbei nicht sauber gearbeitet, können unerwünschte Stauchungen oder Dehnungen bei der Figur entstehen.

Keyframe - Animationen

Um die Animation der Spielfigur zu realisieren, verwenden wir die Keyframe – Animation. Hierzu werden einzelne Schlüsselbilder der Animationen definiert und die Positionen, Größen und die Winkelwerte dieser Bilder werden aufgenommen. Die Bilder zwischen den Schlüsselbildern werden automatisch errechnet. So kann man schon mit wenigen Bildern eine grobe Animation erstellen, welche durch das weitere hinzufügen von Schlüsselbildern immer weiter verfeinert werden kann.

Entwicklung Singleplayer Scripting

Entwicklung Singleplayer Scripting

Unity ist eine Umgebung zum Erstellen von interaktiven 3D Videospielen.

In Unity werden fertig modellierte 3D Objekte importiert und bekommen eine Position im 3D Raum des Spieles. Solche Objekte werden GameObjects genannt. Danach weist man den Objekten Texturen und Collider, welche ermöglichen das die Spielfigur nicht durch Boden durchfällt oder durch Mauern laufen kann zu. Jedem GameObject können ein oder mehrere Scripts zugewiesen werden. Diese Scripts werden in C# oder JavaScript geschrieben.

Die GameObjects können in Unity in einer hierarchischen Struktur angeordnet werden. Das hat den Vorteil, dass beim Bewegen eines Übergeordneten GameObjects das untergeordnete gleich mitbewegt wird. Ein Beispiel wäre ein Auto, wo es ausreichend wäre, die Spielfigur unterzuordnen und nur das Auto zu bewegen. Die Spielfigur würde sich mit dem Auto bewegen.

Ein GameObject kann aus verschiedenen Komponenten bestehen

Mesh

Ein Mesh sind mehrere miteinander verbundene Polygone. Jedem kann ein Material zugewiesen werden.

Material

Ein Material besteht aus einer oder mehreren Texturen.

Rigid Body

Ein Rigid Body ist eine Komponente, die dem GameObject erlaubt, mit Unitys Physik Engine zu interagieren. Mit einem Rigid Body können physikalische Gesetze, wie zum Beispiel die Schwerkraft, simuliert werden.

Collider

Ein Collider ist ein unsichtbares GameObject, welches beim Auftreffen auf einen anderen Collider eine Kollision auslöst.

Trigger

Ein Trigger ist ein Collider der nicht auf Kollisionen reagiert, sondern bei einer Berührung mit einem anderen Collider eine Funktion im Script aufruft.

Hauptobjekte:

Kamera

Die virtuelle Kamera ist auch ein GameObject, welches durch ein Kamera Script gesteuert wird. Das Kamera Script weist dem Kamera GameObject immer eine neue Position zu, welche durch die Position der Spielfigur berechnet wird.

Spielfigur

Die Spielfigur ist ein GameObject mit fertigen Animationen, welches mit einem Script je nach gedrückter Taste in die entsprechende Richtung bewegt wird. Bei jeder Bewegung wird eine bestimmte Animation abgespielt.

Google Analytics:

Durch das Verwenden von Google Analytics, einer Funktion von Google zur Webanalyse, ist es uns möglich das Benutzerverhalten genau zu analysieren. So können wir sehen, wie viele Benutzer pro Tag auf unsere Seite zugreifen. Außerdem haben wir Google Analytics in bestimmte Funktionen im Spiel integriert. So erhalten wir genaue Werte, wie oft ein Spiel begonnen wurde, oder welcher Level am meisten gespielt wurde und wie viele Spieler es ins Ziel geschafft haben. Durch kleine Änderungen könnten wir die Funktionen erweitern, und detailliertere Werte erhalten. Zum Beispiel die Aufenthaltsdauer des Spielers in bestimmten Bereichen des Spiels, oder den genauen Bewegungsverlauf des Spielers im Spiel. Diese Daten sind Statistisch aufgelistet und einem Bestimmten Benutzer NICHT zuordenbar.

Highscores:

Der Spieler kann am Ziel jedes Levels seine Spielzeit in eine für andere Spieler sichtbare Highscore-Liste eintragen und sich somit mit anderen Spielern vergleichen. Die Highscores werden durch das aufrufen einer PHP Seite am Server in eine Datenbank gespeichert. Das Auslesen der Highscores erfolgt ebenfalls über eine PHP Seite welches sich auch auf dem Server befindet.

Fehlerberichte:

Der Spieler hat die Möglichkeit im Spiel ein Formular auszufüllen und uns über Fehler oder andere Wünsche/Beschwerden zu informieren. Alle Berichte werden in einer Datenbank am Server gespeichert und außerdem an unsere Facebook Gruppe gesendet. So haben wir die Möglichkeit schnell zu reagieren und einen Fehler in kürzester Zeit zu beheben.

Marketing

Das Spiel an sich wird von uns kostenlos zur Verfügung gestellt, jedoch haben wir einen anderen Weg gefunden den wirtschaftlichen Faktor hervorzuheben. Die Antwort ist simpel: Werbung. Anfangs werden wir versuchen eine Community mittels einer Facebook Seite und einem Trailer Video welches wir auf YouTube online stellen aufzubauen. Der zweite Schritt ist es, den Link zu unserer virtuellen Schulführung gut erkennbar auf der Schulhomepage zu platzieren. In dem Spiel sind explizit Werbeflächen eingebaut. Während der Benutzer spielerisch die Schule erkundet, entdeckt er gleichzeitig die Werbeflächen. Diese bieten wir Unternehmen an, um zusätzlich zum spielerischen und informativen Faktor auch den finanziellen Faktor in Hinsicht von MicrosizeMe abzudecken.

Entwicklung Multiplayer

Die Idee hinter dem Multiplayermodus des Spieles MicrosizeMe war, einzelne Komponenten und Ideen des Singleplayermodus weiterzuentwickeln und auf den Multiplayermodus zuzuschneiden. Dabei werden vor allem die Bewegungsscripts aber auch andere Komponenten verwendet. Das Ziel ist einen einfachen Multiplayermodus zu entwickeln, wo sich eine bestimmte Anzahl von Spielern über einen Client, der heruntergeladen wird, zu einem, von einer anderen Person, erstellten Spiel zu verbinden und zusammen bzw. gegeneinander zu spielen.

Der Multiplayermodus wird, wie auch das Hauptspiel, die Führung durch die Schule, in Unity designed und die Scripts. Die einzelnen Levels, sowie die anderen Modelle die verwendet werden (z.B. die Spielercharaktere), werden in Cinema 4D modelliert und texturiert.

Funktionen die über das Netzwerk aufgerufen werden, werden RPCs – also Remote Procedure Calls genannt. Das bezeichnet wörtlich eine Funktion die remote, also über einen anderen Client, aufgerufen wird. Diese Funktionen werden bei unserem Multiplayer vor allem verwendet um Mitteilungen, zum Beispiel falls ein Team einen Punkt erzielt an alle Spieler mitzuteilen. Außerdem werden alle Bewegungseingaben an alle Spieler gesendet damit sie die Bewegungen der jeweils anderen Spieler sehen.

Da diese Kommunikation mit den anderen Clients sogenannte „Lags“, also Verzögerungen der Eingaben, die ja zum jedem anderen Client gesendet werden müssen, erzeugen, verwenden wir Extrapolation und Intrapolation. Diese Methoden ermöglichen eine Verringerung der Verzögerungen, da die Bewegungen zwar an alle Clients gesendet werden – allerdings werden die Bewegungen schon im Voraus auf jedem Client extra berechnet. Das heißt wenn ein Spieler vorwärts geht, wird das an alle Clients gesendet und angenommen das der Spieler weiter vorwärts geht, also die andere Spielfigur auf dem eigenen Computer schon weiterbewegt wird während auf weitere eingaben des eigentlichen Spielers gewartet wird. Falls der Spieler jetzt aber nicht weiter vorwärts geht, sondern beispielsweise zur Seite wird die nächste Position nicht mehr nach vorne berechnet sondern an der Seite. Dadurch entstehen zwar manchmal leicht falsche Positionen (je nach Geschwindigkeit des Netzwerkes) allerdings werden „ruckartige“ Bewegungen reduziert.

Im Multiplayermodus spielt man „capture the flag“, also „erobere die Flagge“, wobei zwei Teams gegeneinander antreten und versuchen eine Flagge, die in der Mitte des Spielfeldes platziert ist, aufzunehmen und in ihre eigene Basis zu tragen, um einen Punkt für ihr eigenes Team zu erzielen. Dabei sollten die einzelnen Levels so aufgebaut werden um beiden Teams dieselben Chancen zu gewähren, aber gleichzeitig nicht auf beiden Hälften genau gleich aussehen. Es ist wichtig, dass beide Teams ungefähr dieselbe Zeit zur Flagge brauchen und ähnliche Hindernisse zu überwinden haben.

Es gibt verschiedene Klassen die jeweils unterschiedliche Fähigkeiten besitzen. Eine Klasse ist eher dafür geeignet die Flagge aufzunehmen und sie in die eigene Basis zu

transportieren. Um ihr dies zu ermöglichen besitzt diese Klasse verschiedene Fähigkeiten. Die erste Fähigkeit ermöglicht einen kurzzeitigen Tempoboost, die zweite Fähigkeit ein Teleport ein paar Schritte nach vorne um aus Gefahrensituationen zu entkommen. Die Andere Klasse ist eher unterstützend und besitzt eine Fähigkeit um Verbündete als Hilfe, bzw. den Gegner um ihn aufzuhalten, an sich zu ziehen. Weiters kann er ein Schutzschild aufstellen, was Gegnern enge Passagen blockiert.

Bericht der Projektkoordinator/In

Zusammenarbeit innerhalb des Teams

Die Zusammenarbeit in unserem Team basiert auf einer stetigen Kommunikation zwischen den Teammitgliedern. Jedes Teammitglied soll immer auf dem aktuellen Stand sein und somit über den Fortschritt der Kollegen informiert sein. Um diesen stetigen Informationsaustausch über schulfreie Tage hinweg aufrecht zu erhalten überlegten wir uns ein System. Abgesehen von den fünf Unterrichtseinheiten, die uns wöchentlich in der Schule zu Verfügung stehen, halten wir uns über Facebook immer auf dem neuesten Stand und helfen uns gegenseitig aus. So konnten wir auch die schwierigsten Probleme, die während der Entwicklung auftraten, gemeinsam lösen. Bei komplizierten Problemen nutzen wir auch den Sprachchat Skype. Auch Treffen bei Teammitgliedern zu Hause sind keine Seltenheit. Das Projekt ist in verschiedene Zuständigkeitsbereiche geteilt, dennoch trägt jeder einzelne die Verantwortung für das gesamte Projekt.

Zusammenarbeit zwischen Team und Projektbetreuer

Die Tatsache dass Prof. Baumgartner unser Team bereits seit drei Jahren unterrichtet erleichtert die Zusammenarbeit um einen erheblichen Faktor. Er schafft es immer uns mit konstruktiver Kritik und Verbesserungsvorschlägen zu helfen und uns auch den nötigen Freiraum für Individualität und Kreativität zu überlassen. Außerdem stimmen die Ansprüche, die das Team und der Projektbetreuer an das Projekt stellen überein. Von Zeit zu Zeit musste er uns auch bremsen, da wir mit unseren Ideen doch zu weit gingen und die Grenzen der Realisierbarkeit überschritten.

Außerdem schließen wir den Projektbetreuer in den Kreis der Kommunikation mit ein, so dass er immer auf dem neuesten Stand ist.

Zusammenarbeit zwischen Team und Schule

Die Schule hilft uns sehr, was die Präsentation unseres Projekts angeht. Im Gegenzug dazu ist unser Projekt Werbung für die Schule. Außerdem hat die Schule die Möglichkeit durch den Erlös der Werbeflächen etwas zu verdienen. Auch im Bereich der Schularchitektur wurde uns sehr geholfen. Von der Schule bekamen wir die Pläne des Schulgebäudes. Eine Verlinkung zu unserer Seite auf der Schulhomepage wurde uns bereits versichert.

Lernen im Team

Das Lernen im Team fiel uns sehr leicht. Da wir die Recherchen alle zusammen betrieben, konnten wir Unklarheiten sofort im Team besprechen. Außerdem hatte ein Teammitglied im Bereich Modellierung bereits viele Erfahrung gesammelt. Das vorhandene Wissen erleichterte die Recherche für die Teamkollegen.

Wie schon erwähnt ist jeder aus dem Team ein leidenschaftlicher Computerspieler. Dadurch konnten wir das in diesem Bereich gesammelte Wissen untereinander austauschen und dabei etwas lernen.

Diskussionen und Konflikte

Im Laufe unseres Projekts gab es bis jetzt keine größeren Konflikte innerhalb des Teams. Diskussion über die Gestaltung des Spiels dafür umso öfter. Oft waren wir uns nicht einig, wie wir einzelne Dinge realisieren sollten oder ob gewisse Features aus zeitlichen Gründen überhaupt den Weg ins Spiel finden sollten. Jedoch kamen wir bis jetzt immer zu einem Kompromiss mit dem alle leben konnten und welche wahrscheinlich das Beste für das Projekt gewesen sind

Anhang

Abbildung 1: 3D-Modell des Tronbikes ohne Texturen

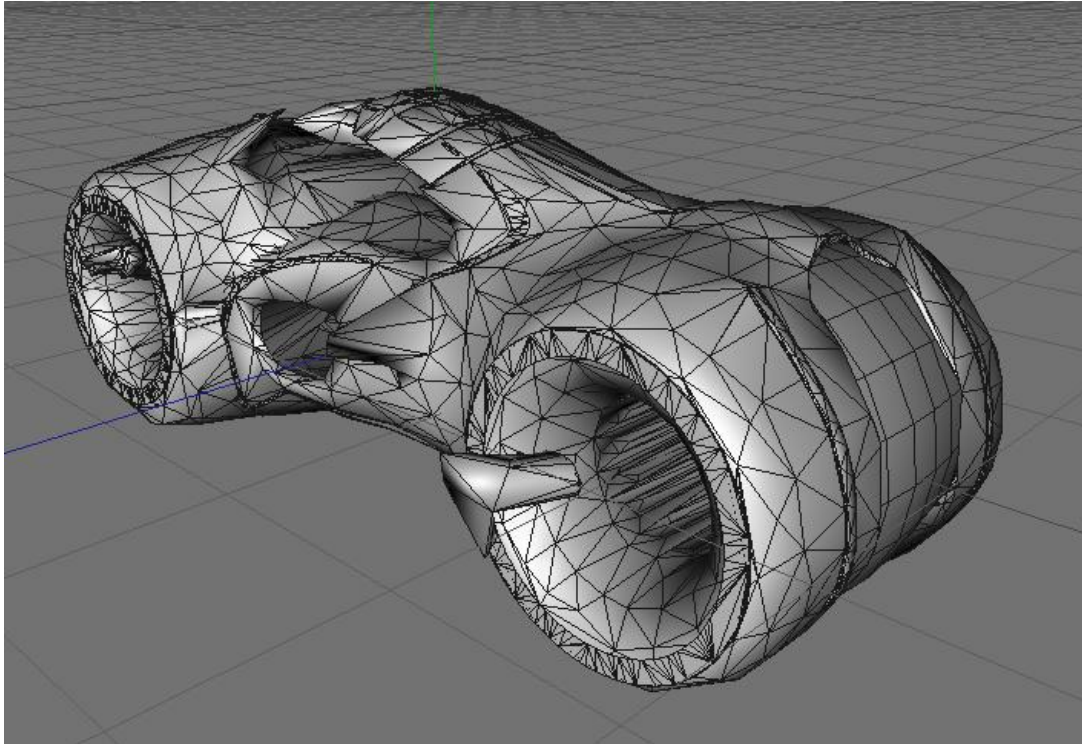


Abbildung 2: Als Bild abgespeicherte Texturen für das Bike

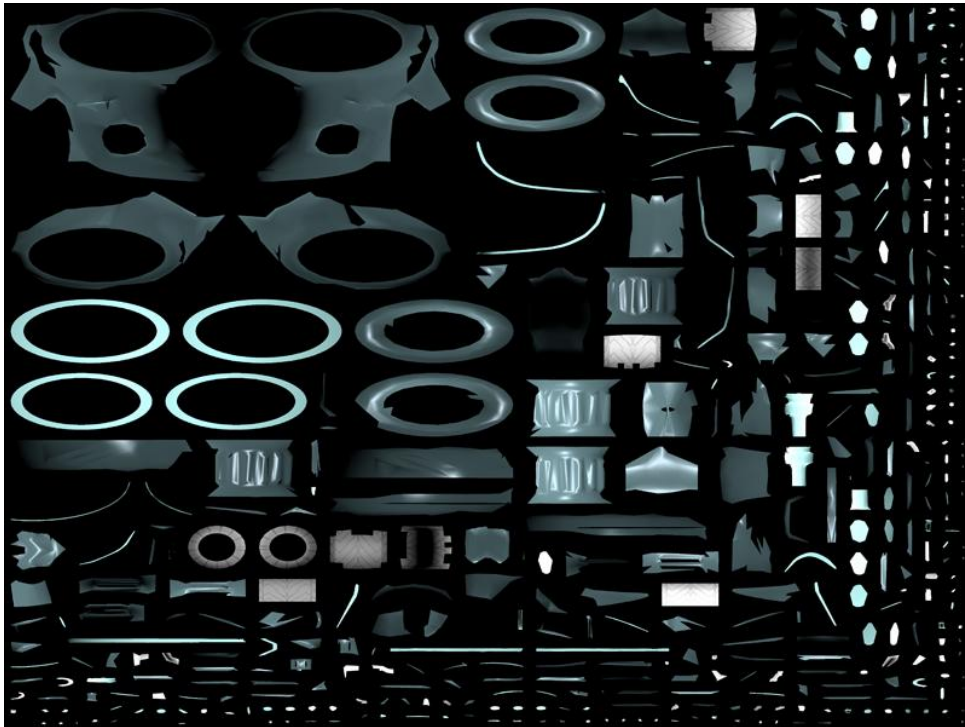


Abbildung 3: Fertig texturiertes Tronbike



Abbildung 4: Stark verkleinertes Panoramabild zum Außenbereich der Schule



Abbildung 5: Sky-Objekt mit realistischer Darstellung des bewölkten Himmels



Abbildung 6: Katapult, das den Spieler auf höher gelegene Plattformen befördert

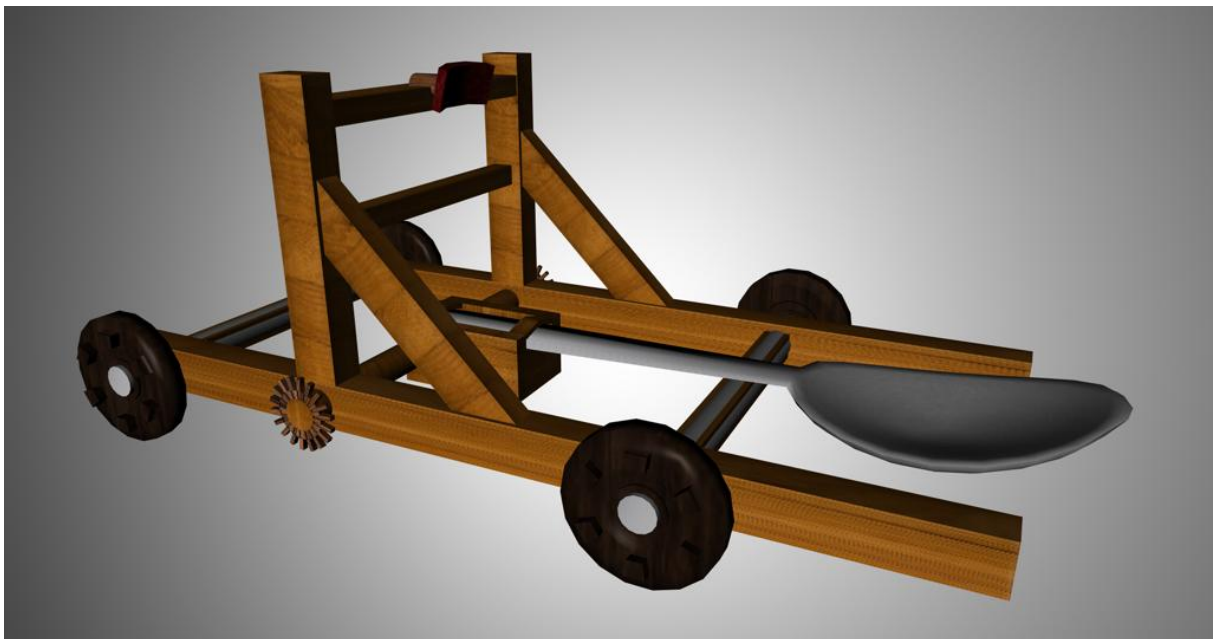


Abbildung 7: Vom Geländer herunterrollende Getränkedosen



Abbildung 8: Der zweite Level des Computerspiels

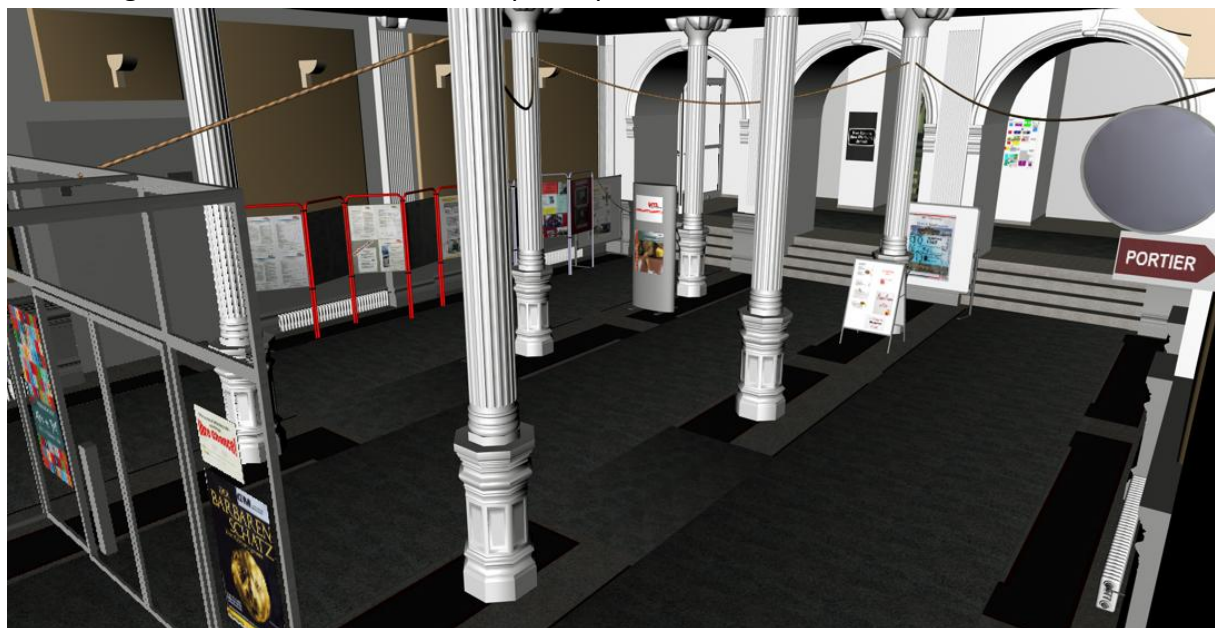


Abbildung 9: Detailliert nachmodellerte Säule bestehend aus ungefähr 20.000 Referenzpunkten

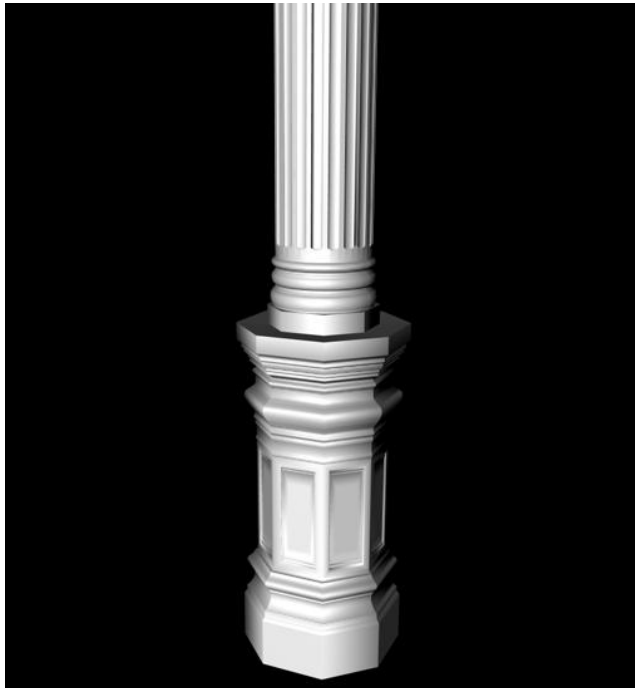


Abbildung 10: Keyframe 337

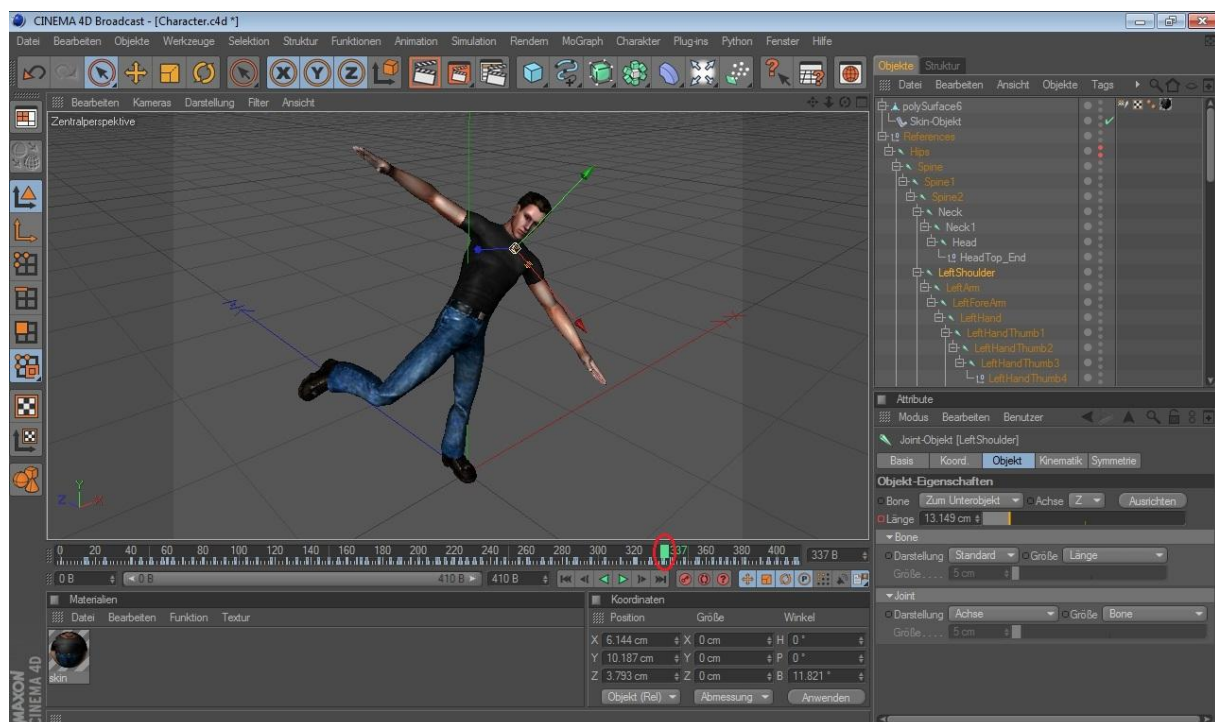


Abbildung 11: Keyframe 341

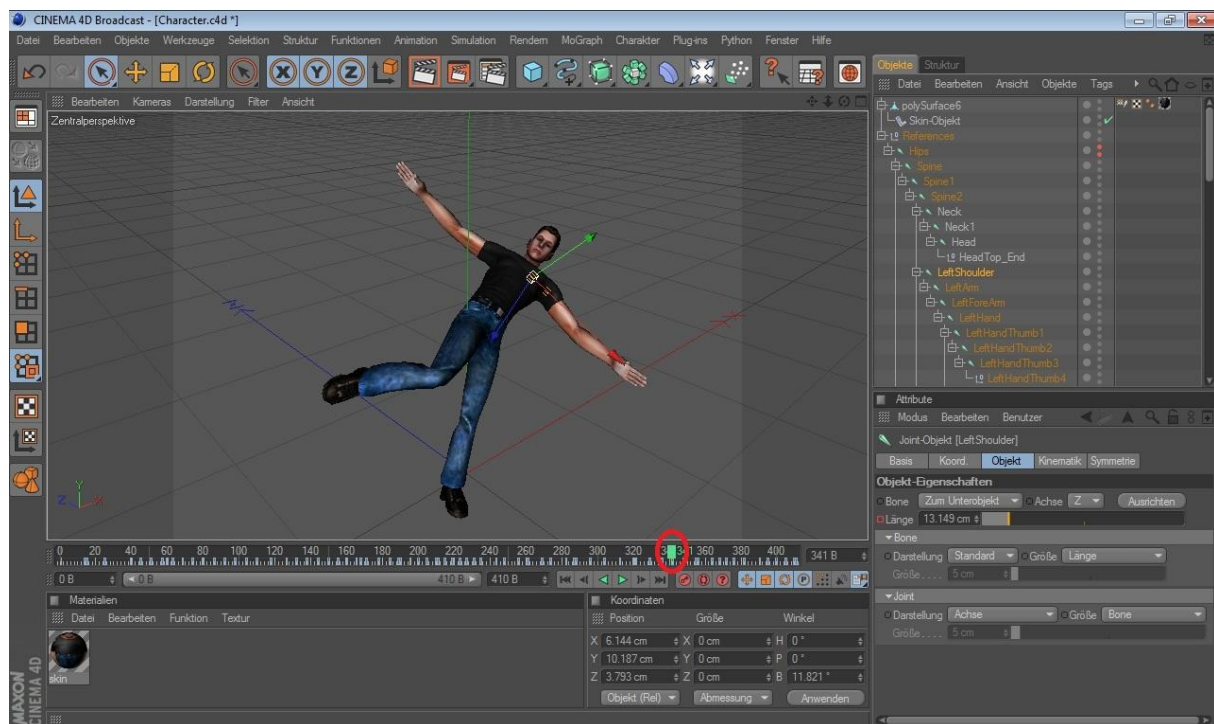


Abbildung 12: Ansicht während der Gewichtung. In Rot kann man die Bones der Spielfigur über das ganze Modell verteilt sehen. Die rote Fläche zeigt die Gewichtung des momentan angewählten Joints.

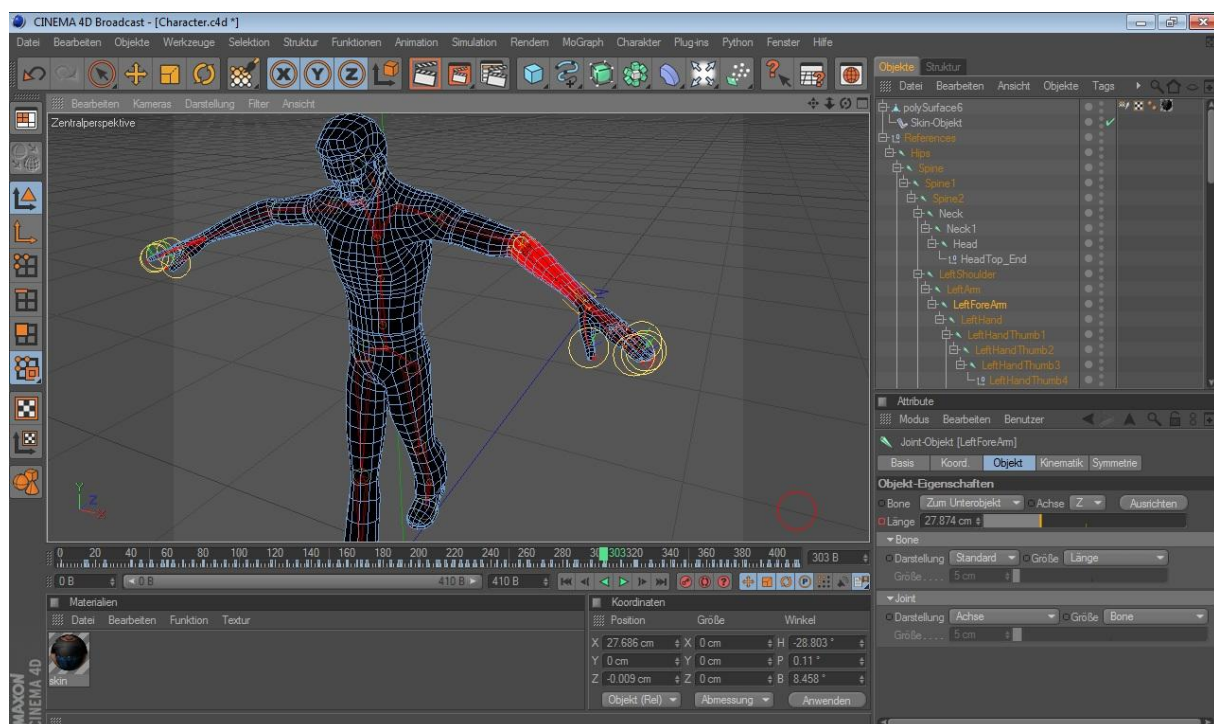


Abbildung 13: Unity Interface

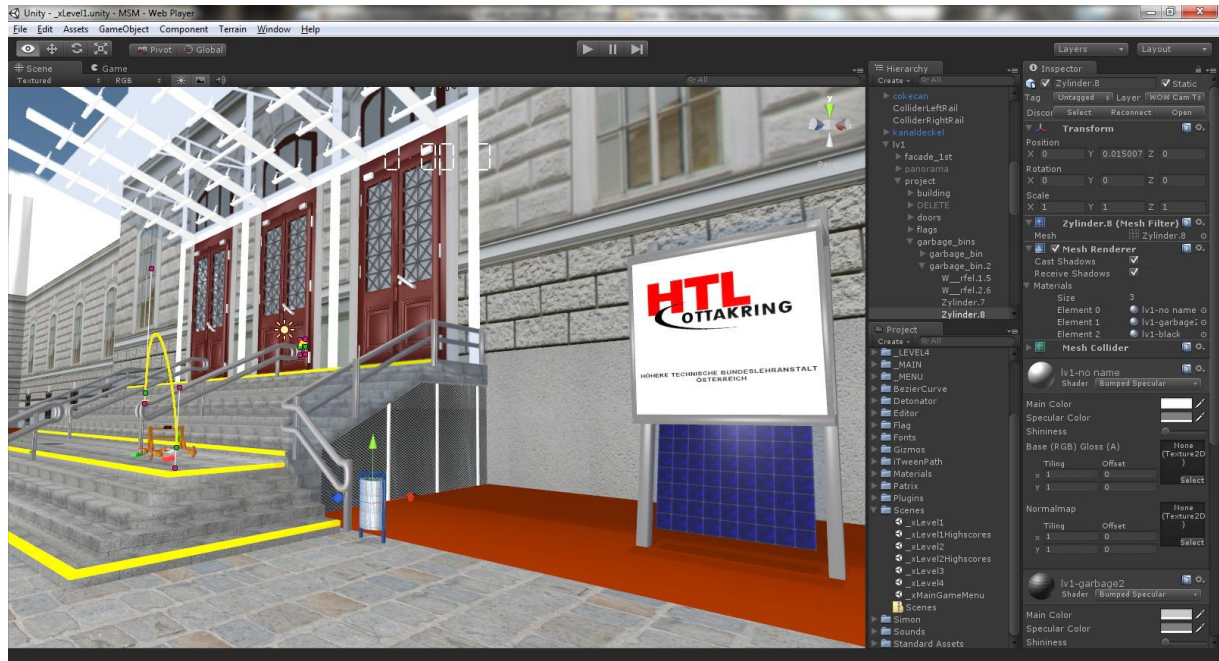


Abbildung 14: Mesh Collider GameObject

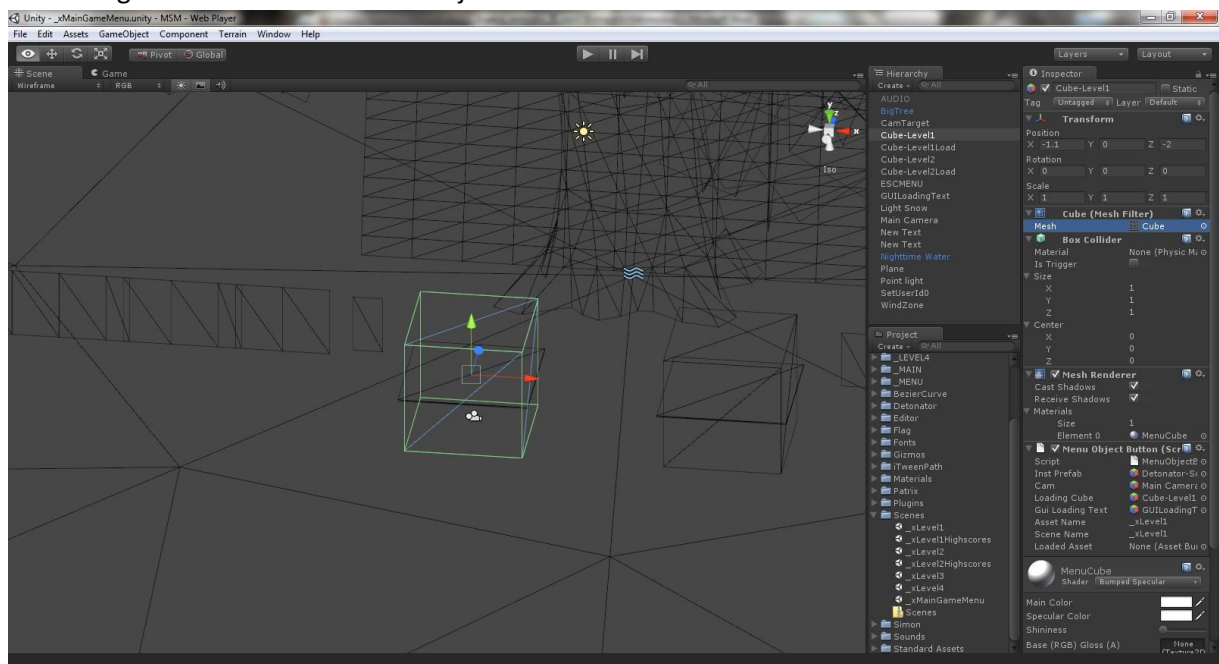


Abbildung 15: Kamera-Ansicht in Unity 3D

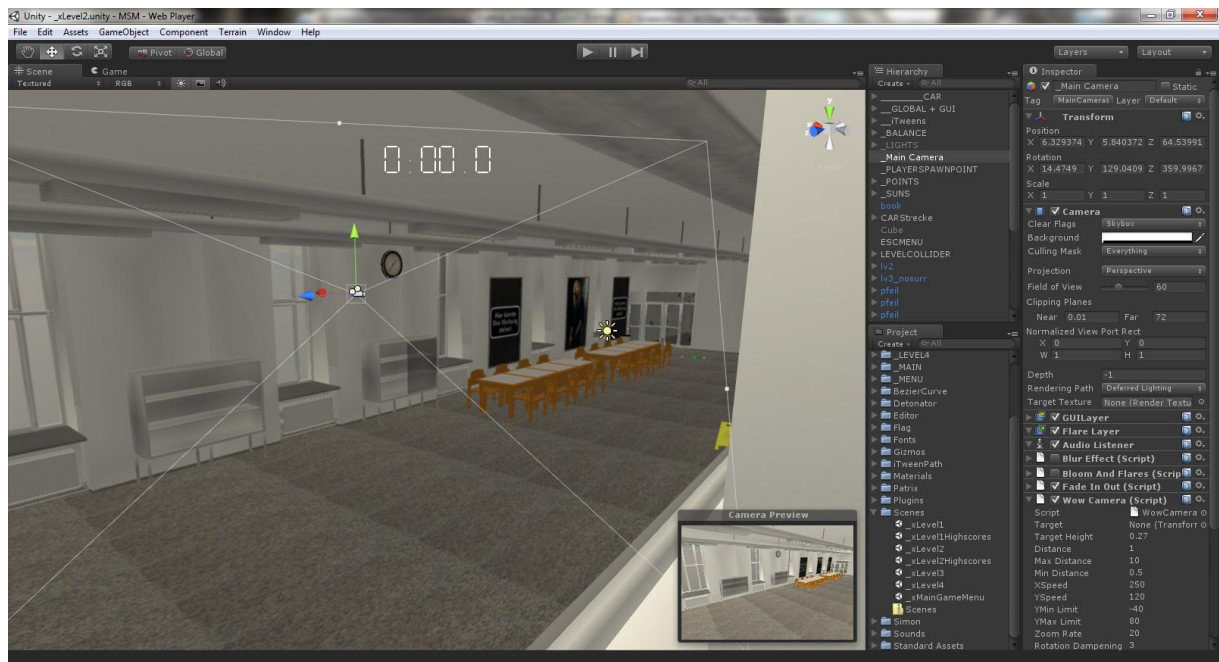


Abbildung 16: Charakter mit eingblendetem Collider

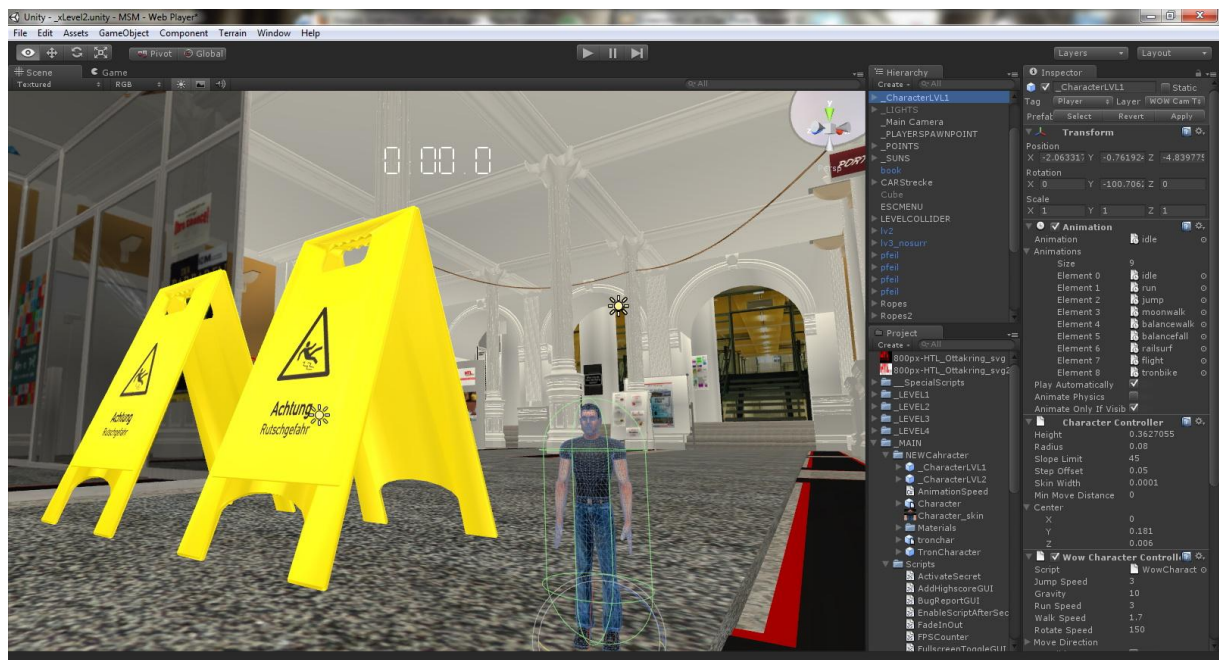


Abbildung 17: Statistische Daten mittels Google Analytics aufgezeichnet und ausgewertet

